

CLAIMS

Please amend the Claims as follows:

1. (Currently Amended) A method for load balancing in a tightly-coupled multiprocessor computer system comprising ~~the steps of:~~
 - dividing a task into a plurality of subtasks;
 - placing the plurality of subtasks into a centralized task queue;
 - distributing the plurality of subtasks in the centralized task queue to a plurality of library processors, wherein each library processor comprises exactly two task buffers;
 - wherein at least one subtask from the plurality of subtasks in the centralized task queue is distributed to at least one of the plurality of library processors when the library processor has at least one empty task buffer; and
 - wherein distributing a subtask from the plurality of tasks in the centralized task queue to the one of the plurality of library processors comprises the one of the plurality of library processors fetching the subtask from the centralized task queue.
2. (Previously Presented) The method of claim 1, further comprising distributing the subtask from the plurality of subtasks in the centralized task queue to the one of the plurality of library processors when the one of the plurality of library processors has one or two empty task buffers.
3. – 4. (Cancelled).
5. (Previously Presented) The method of claim 1, further comprising distributing the subtask from the plurality of subtasks in the centralized task queue to the one of the plurality of library processors by the one of the plurality of library processors fetching the subtask from the centralized task queue when the load of the one of a plurality of library processors is zero or one subtasks.
6. (Previously Presented) The method of claim 1, further comprising distributing the subtask from the plurality of subtasks in the centralized task queue to the one of the plurality of library

processors by the one of the plurality of library processors fetching the subtask from the centralized task queue when the load of the one of a plurality of library processors is zero subtasks.

7. – 8. (Cancelled).

9. (Previously Presented) A system for load balancing in a tightly-coupled multiprocessor computer system comprising:

- a system kernel configured to receive a task and to divide the received task into a plurality of subtasks;

- a library task queue coupled to the system kernel;

- a plurality of library processors coupled to the library task queue, wherein each of the plurality of library processors comprises exactly two task buffers;

- wherein the system kernel is configured to place the plurality of subtasks into the library task queue; and

- wherein each of the plurality of library processors is configured to fetch at least one subtask of the plurality of subtasks from the library task queue.

10. (Previously Presented) The system of Claim 9, wherein each of the plurality of library processors further is further configured to fetch a subtask from the library task queue when that library processor has at least one empty task buffer.

11. (Cancelled).

12. (Previously Presented) The system of Claim 9, wherein the system kernel is comprised of a single processor.

13. (Previously Presented) The system of Claim 9, wherein the system kernel is comprised of a plurality of processors.

14. (Cancelled).

15. (Previously Presented) A computer program product for load balancing in a tightly-coupled multiprocessor computer system, the computer program product having a medium with a computer program embodied thereon, the computer program comprising:

computer code for dividing a task into a plurality of subtasks;

computer code for placing the plurality of subtasks into a centralized task queue;

computer code for distributing the plurality of subtasks in the centralized task queue to a plurality of library processors, wherein each library processor comprises exactly two task buffers;

wherein a subtask from the plurality of subtasks in the centralized task queue is distributed to one of the plurality of library processors when the library processor has at least one empty task buffer; and

wherein distributing a subtask from the plurality of subtasks in the centralized task queue to the one of the plurality of library processors comprises the one of the plurality of library processors fetching the subtask from the centralized task queue.

16. (Previously Presented) The computer program product of Claim 15, further comprising computer code for distributing the subtask from the plurality of subtasks in the centralized task queue to the one of the plurality of library processors when the one of the plurality of library processors has one or two empty task buffers.

17. – 18. (Cancelled).

19. (Previously Presented) The computer program code of Claim 15, further comprising computer code for distributing the subtask from the plurality of subtasks in the centralized task queue to the one of the plurality of library processors by the one of the plurality of library processors fetching the subtask from the centralized task queue when the load of the one of a plurality of library processors is zero or one subtasks.

20. (Previously Presented) The computer program code of Claim 15, further comprising computer code for distributing the subtask from the plurality of subtasks in the centralized task queue to the one of the plurality of library processors by the one of the plurality of library

processors fetching the subtask from the centralized task queue when the load of the one of a plurality of library processors is zero subtasks.

21. – 22. (Cancelled).